

An Adjusted Recursive Operator Allocation Optimization Algorithm for Line Balancing Control

B.L. Song, W.K. Wong, J. Fan, and S.F. Chan¹

Abstract—This paper aims to solve the operator allocation optimization problem for line balancing control under two unsatisfied conditions. An approach is proposed for combination condition adjustment. An adjusted recursive operator allocation optimization algorithm is developed for generating the optimal solution under these two conditions adjusted.

Index Terms—Operator allocation, Optimization, Recursive algorithm, Assembly line balancing, and Operator efficiency

I. INTRODUCTION

In the apparel industry, the planning and line-balancing decisions are heavily relied on the production experts. However, industrial experience shows that it is difficult for them to achieve a perfectly balanced line so that each operation can keep the same production rate [1]. It is because experts make decisions on the basis of their experience. The decisions are thus not scientific or optimal [2]. The complexity of combinatorial optimization caused by operator's multiple skills and variant efficiency on different operations make the matter even worse for optimal production control [3].

The development of an operator allocation optimization algorithm in apparel manufacture based on operator efficiency prediction is therefore significant in achieving the optimal solution to support the expert's decision making on line-balancing control [4]. An operator efficiency prediction (OEP) approach has been proposed with the data collected by UPS system [5]. A recursive operator allocation optimization algorithm has been developed for the situation that the combination condition is satisfied [6].

This paper is dedicated to solve the operator allocation optimization problem for two other situations where combination condition is not satisfied. The adjustment of combination condition is presented and the adjusted operator allocation optimization algorithm is further developed.

This paper is organized as follows: following the introduction, section 2 demonstrates the flow of the adjusted recursive operator allocation optimization, and explains two combination conditions that need to be adjusted. Section 3

presents the approach of adjustment. Section 4 introduces briefly about the adjusted operator allocation optimization algorithm, which includes recursive combination algorithm, recursive operator allocation algorithm and optimization, under these two situations. Section 5 conducts an experiment and reports the results. A conclusion is made in Section 6.

II. PROBLEM OF ADJUSTED OPERATOR ALLOCATION OPTIMIZATION

A. Nomenclature

The nomenclature used in this paper is summarized as follows.

S_{opr} = the set of all operators,

S_{opt} = the set of all operations,

N_{sopr} = the total number of all operators,

N_{sopt} = the total number of all operations,

N_{sopr} = the total number of operators that are really needed,

$SAM[i]$ = the standard allowed minutes to finish a fixed quantity garments in the i^{th} operation (e.g. 100 pieces),

T_{SAM} = the total SAM of all operations,

$S_{opr}[i]$ = the set of all operators available for operation i ,

$N_{sopr}[i]$ = the number of operators available for operation i ,

$S_{sopr}[i]$ = the set of single-skilled operators available for operation i ,

$N_{S_{sopr}}[i]$ = the number of single skilled operators available for operation i ,

$S_{mopr}[i]$ = the set of multiple-skilled operators available for operation i ,

$N_{S_{mopr}}[i]$ = the number of multiple-skilled operators available for operation i ,

$S_{nopr}[i]$ = the set of operators really needed for operation i in a new production order,

$N_{S_{nopr}}[i]$ = the number of operators really needed on operation i ,

$S_{nmopr}[i]$ = the set of multiple-skilled operators really needed for operation i ,

$N_{S_{nmopr}}[i]$ = the number of multiple-skilled operators really needed for operation i ,

¹ This work was fully supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region, China. (Project No. Polyu 5288/03E)

B.L. Song, W.K. Wong, J. Fan, and S.F. Chan. are with the Institute of Textiles and Clothing, the Hong Kong Polytechnic University, Hong Kong. (corresponding author: B.L. Song; phone: 852-27666453; fax: 852-27731432; e-mail: tcsongbl@inet.polyu.edu.hk).

$E_{opr}[i, j]$ = the j^{th} operator's efficiency on the i^{th} operation, and
 $E_{sopr}[i, j]$ = the j^{th} single skilled operator's efficiency on the i^{th} operation.

B. Assumptions and Problem Description

Constraints of the operator allocation for a single model stochastic (SMS) assembly line to be studied are described as follows: A set of operators will be allocated to do a new job on a hybrid Unit Production System (UPS), where the operation process sequence is serial, while in a particular operation operators operate in parallel order. Each operator will only work on one machine at a time. There is no precedence order among operators working on a same operation. Operation processing time is stochastic. Each operator has his/her own skill matrix. An operator who is able to do more than one operation is called a multiple-skilled operator, while an operator who is able to do only one operation is called a single-skilled operator. An operator's efficiency varies on different operations.

To formulate the problem, the set of all operators is expressed as the set of single-skilled operators plus the set of multiple-skilled operators. The numbers of elements in these sets have the same relationship as (1).

$$S_{opr}[i] = S_{sopr}[i] \text{ set plus } S_{mopr}[i], \quad (1)$$

$$N_{sopr}[i] = N_{S_{sopr}}[i] + N_{S_{mopr}}[i] \quad (2)$$

The theoretic optimal needed operator number for each operation ($N_{S_{opr}}[i]$) can be derived by considering both the Standard Allowed Minutes (SAM) of each operation, and the total number of available operators ($N_{S_{opr}}$). As single skilled operators have limited allocation flexibility, they are granted the priority to be allocated to the operations they can do first in order to reduce combinatorial complexity.

A feasible operator allocation can only be achieved when the combinatorial condition in (3) is satisfied, that is, on any operation, the number of operators really needed should be larger than that of single skilled operators, and smaller than that of total available operators.

$$N_{sopr}[i] \geq N_{S_{opr}}[i] \geq N_{S_{sopr}}[i] \quad (3)$$

However, during production, two other situations may occur, where operators really needed are less than single skilled operators as in (4), or more than total available operators as in (5). In those cases, conditions should be adjusted to meet (3).

$$N_{S_{opr}}[i] < N_{S_{sopr}}[i], \quad (4)$$

$$N_{sopr}[i] < N_{S_{opr}}[i] \quad (5)$$

Condition adjustment will be conducted under four assumptions: 1. No more operators will be available for this order. 2. To guarantee production capacity, keep as more operators as possible. 3. If some operators have to be removed,

operator efficiency on a particular operation is the criteria to choose whom to leave. The removed operator will be considered to serve other lines. 4. To keep the assembly line optimally balanced is the primary objective in this paper.

In conclusion, the problem in this paper is to discuss how to adjust conditions in (4) and (5) to objective condition in (3) and how to achieve an optimal operator allocation under these two adjusted conditions.

C. Flow of Adjusted Recursive Operator Allocation Optimization

Assigning the appropriate operators to the appropriate operations can achieve a balanced assembly line with the highest production line efficiency and lowest operator efficiency waste [6]. The number of operators allocated to each operation is determined by the theoretic optimal needed operator number for each operation. As each single skilled operator can only be allocated to one particular operation he can do, problem complexity is mainly caused by operators with multiple skill matrix. A recursive optimization algorithm is developed to achieve the optimal allocation if combination is possible. The flow of recursive operator allocation optimization is demonstrated in Fig 1.

The combination condition must be met for each operation before generating a feasible operator allocation. On any operation i , if the number of single skilled operators available is more than really needed, some single skilled operators will be removed to make the condition in (6) satisfied; otherwise if the number of total operators available on operation i is less than really needed, needed operators will be adjusted till condition in (7) is satisfied. This condition inspection will be conducted in a loop, until the condition for all the operations are satisfied. This condition inspection and adjustment loop is presented by Fig.2.

$$N_{S_{opr}}[i] \geq N_{S_{sopr}}[i] \quad (6)$$

$$N_{S_{opr}}[i] \geq N_{S_{opr}}[i] \quad (7)$$

III. ADJUSTMENT OF CONDITIONS

There are more than one adjustment solutions. This section will demonstrate how to make an optimal adjustment for two conditions respectively with the consideration of both the objectives of (6) and (7) and previous assumptions.

A. Adjustment for condition of $N_{S_{opr}}[i] < N_{S_{sopr}}[i]$

If condition (4) occurs on the i^{th} operation, some single skilled operators will be removed from total available operators. We need to determine the optimal solution of whom and how many of them are going to be removed.

Let X represent the number of single skilled operators to be removed. R is the ratio of SAM of the i^{th} operation over total SAM of all operations.

$$R = SAM[i] / T_{SAM} \quad (8)$$

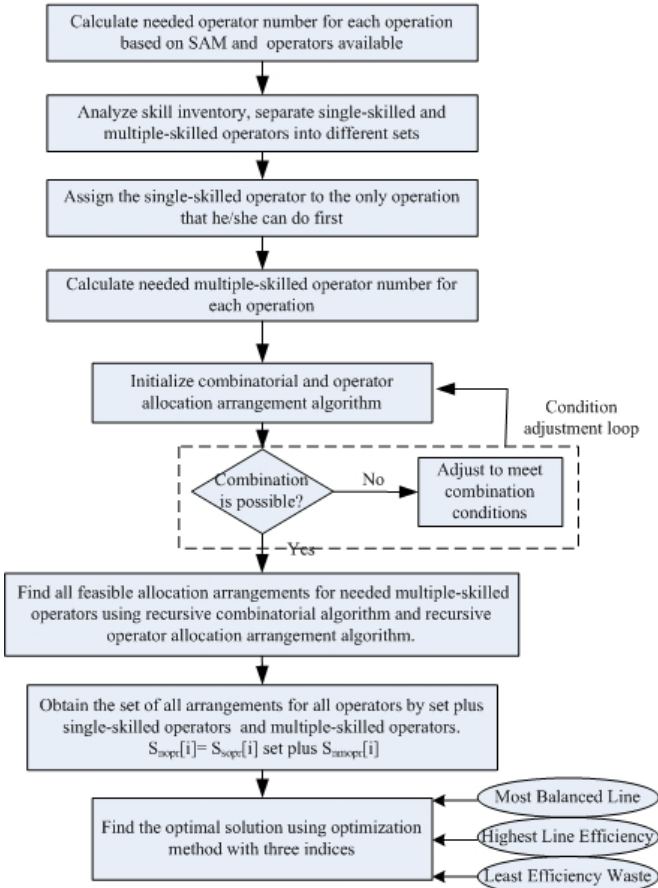


Fig. 1. Flow chart of recursive operator allocation solution

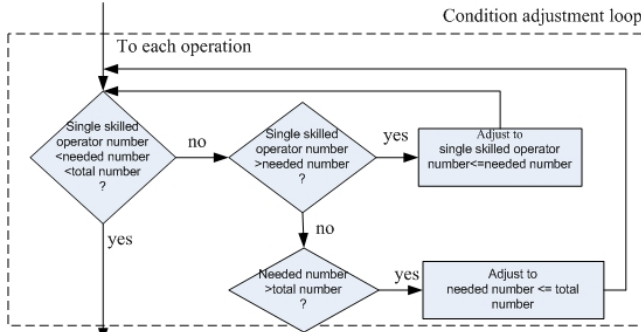


Fig. 2. Flow chart of condition adjustment

The number of single skilled operators left for the i^{th} operation changes to (9)

$$N_{S_{opr}}[i] = N_{S_{opr}}[i] - X \quad (9)$$

The number of total operators left changed to (10)

$$N_{opr}[i] = N_{opr}[i] - X \quad (10)$$

The theoretic needed operators for the i^{th} operation is calculated by (11)

$$N_{S_{opr}}[i] = (N_{S_{opr}}[i] - X) * R \quad (11)$$

To meet the objective of (6), we should let

$$(N_{S_{opr}}[i] - X) * R \geq N_{S_{opr}}[i] - X \quad (12)$$

Then we will have:

$$X \geq \frac{N_{S_{opr}}[i] - N_{S_{opr}}[i] * R}{1 - R} \quad (13)$$

As we assume to keep as many operators as possible, that is, to maximize (9), correspondingly we should minimize X . The optimal value of integer of X is therefore set as (14)

$$X = \text{Trunc}\left(\frac{N_{S_{opr}}[i] - N_{S_{opr}}[i] * R}{1 - R}\right) \quad (14)$$

According to assumption 3, operator efficiency on a particular operation is the criteria to determine whom to be chosen. The efficiency on each operation of all operators in single skilled operator set ($S_{opr}[i]$) will be sorted in an increasing order. The operator with the lowest efficiency has the highest possibility to be removed. Thus the first X operators in $S_{opr}[i]$ will be taken out from operating this order. The total number of operators is reduced. So the theoretic needed operators for the other operations should be recalculated accordingly.

B. Adjustment for condition of $N_{S_{opr}}[i] < N_{S_{opr}}[i]$

Under condition (5), total available operators on the i^{th} operation are less than operators really needed. As there are no more operators who can be added for the allocation as assumption 1, we have to reduce the number of needed operators till objective of (7) can be met, that is

$$N_{S_{opr}}[i] \leq N_{S_{opr}}[i] \quad (15)$$

However, we should also abide by the assumption 2 to keep as many operators as possible, which is to maximize $N_{S_{opr}}[i]$.

Hence the optimal solution is to set $N_{S_{opr}}[i] = N_{S_{opr}}[i]$ (16)

The number of needed operators for each operation should follow a fixed proportion [6]. Since one of them changes, all the others should be recalculated in proportion. The total number of operators available N_{opr} keeps unchanged, while the number of total needed operators becomes less. That who are selected and who are removed is not determined by the adjustment condition but by the operator allocation optimization algorithm. There may be more than one solution to be generated. The optimal one will be taken as the final solution.

IV. ALGORITHMS OF ADJUSTED RECURSIVE OPERATOR ALLOCATION OPTIMIZATION

A. Recursive Operator Allocation Optimization Algorithm

All feasible allocations of needed multiple-skilled operators will be generated by employing the concept of recursion [6].

Given the number of operations N_{opr} (simplified as n), A_i represents a particular operator allocation for the i^{th} operation. $A_{i \rightarrow n}$ denotes a particular operator allocation from the first operation to the i^{th} operation. S_A denotes a set of operator allocations of A . $a_{s,i}$ is the i^{th} element in set S . $N(S)$ is

the number of elements in set S . At the beginning, as there is no operator allocation, set $i=0$, $A_{i \rightarrow i} = \Phi$. Given $A_{i \rightarrow i}$, we continue to find the set of all possible operator allocations for all operations $S_{A_i \rightarrow n | A_{i \rightarrow i}}$. The recursive definition to obtain $S_{A_1 \rightarrow n | A_{1 \rightarrow i}}$ is presented in (17).

$$S_{A_1 \rightarrow n | A_{1 \rightarrow i}} = \begin{cases} \{A_{1 \rightarrow i}\} & i=n \\ \left\{ \begin{array}{l} S_{A_1 \rightarrow n | A_{1 \rightarrow i-1}} \cup S_{A_1 \rightarrow n | A_{1 \rightarrow i-1}}^{+a} \\ S_{A_1 \rightarrow n | A_{1 \rightarrow i-1}} \cup S_{A_1 \rightarrow n | A_{1 \rightarrow i-1}}^{+a_2} \\ \dots \\ S_{A_1 \rightarrow n | A_{1 \rightarrow i-1}} \cup S_{A_1 \rightarrow n | A_{1 \rightarrow i-1}}^{+a_m} \end{array} \right\} & i < n \text{ and } N(S_{A_1 \rightarrow n | A_{1 \rightarrow i-1}}) > C \\ \{\Phi\} & i < n \text{ and } N(S_{A_1 \rightarrow n | A_{1 \rightarrow i-1}}) = C \end{cases} \quad (17)$$

B. Recursive Combinatorial Algorithm

A general algorithm for obtaining the set of all combinations of any m elements from a set S is proposed as follows. It is used in the above operator allocation algorithm to generate the set of all possible combinations of needed multiple-skilled operators for a specific operation.

Given $S = \{a_{s,1}, a_{s,2}, \dots, a_{s,N(S)}\}$. $S_{i \rightarrow N(S)}$ represents the subset of S including the elements from $a_{s,i}$ to $a_{s,N(S)}$ continuously, that is: $\{a_{s,i}, a_{s,i+1}, \dots, a_{s,N(S)}\}$, $a_{s,i} \cup a_{s,j}$ is the union operation of the element $a_{s,i}$ with $a_{s,j}$. $CS_{i \rightarrow N(S)}^m$ is the set of all combinations of any m elements from the set S . If given $S_{i \rightarrow N(S)}$, $CS_{i \rightarrow N(S)}^m$ can be obtained by (18).

$$CS_{i \rightarrow N(S)}^m = \begin{cases} \{\Phi\} & N(S_{i \rightarrow N(S)}) < m \text{ or } m=0 \\ \{a_{s,i} \cup a_{s,i+1} \cup \dots \cup a_{s,N(S)}\} & N(S_{i \rightarrow N(S)}) = m \\ \left\{ \begin{array}{l} a_{s,i}^a C_{i+1-N(S)}^{m-1}, a_{s,i+1}^a C_{i+1-N(S)}^{m-1}, \dots, a_{s,i}^a C_{i+1-N(S)}^{m-1} N(C_{i+1-N(S)}^{m-1}) \\ a_{s,i+1}^a C_{i+2-N(S)}^{m-1}, a_{s,i+2}^a C_{i+2-N(S)}^{m-1}, \dots, a_{s,i+1}^a C_{i+2-N(S)}^{m-1} N(C_{i+2-N(S)}^{m-1}) \\ \dots \\ a_{s,N(S)}^a C_{i+1-N(S)}^{m-1} \end{array} \right\} & N(S_{i \rightarrow N(S)}) > m \end{cases} \quad (18)$$

C. Optimization

Three optimal indices, namely, efficiency of bottleneck operation (Eff of bottleneck), standard deviation of operation efficiency (Std of opreff), and total operation efficiency waste (total opreff waste), are proposed to evaluate the goodness of a specific operator allocation so as to find the optimal solution.

V. EXPERIMENT AND RESULTS

A. Case of Adjustment for Condition of $N_{S_{opr}[i]} < N_{S_{opr}[i]}$

The following experiment will demonstrate with case 1 the

approach of how to adjust the condition of $N_{S_{opr}[i]} < N_{S_{opr}[i]}$. The result of running the adjusted operator allocation optimization algorithm is also reported.

In this case, before adjustment, the set of all operators S_{opr} is {Opr1, Opr2, Opr3, Opr4, Opr5, Opr6, Opr7, Opr8}, the set of all operations S_{opr} is {Opt1, Opt2, Opt3}, total number of operators $N_{S_{opr}}$ is 8 and total number of operations $N_{S_{opr}}$ is 3. Skill matrix of each operator ($S_{opr}[i]$, $S_{opr}[i]$, $S_{opr}[i]$), and predicted efficiency of operator j on operation i ($E_{opr}[i, j]$) are shown on left side of table 1 respectively. Based on that SAM = {1.20, 1.60, 1.80} and total operator number is 8, the number set of operators needed for each operation $N_{S_{opr}[i]}$ is calculated as {3, 1, 4}. The number of total available operators for each operation $N_{S_{opr}[i]}$ is {3,4,4} and the number of single skilled operators for each operation $N_{S_{opr}[i]}$ is {1,2,2}.

Table 1: Condition adjustment for operator allocation in case 1

$i \backslash j$	Before adjustment			After adjustment		
	Opt1	Opt2	Opt3	Opt1	Opt2	Opt3
S_{opr}	Opr1, Opr2, Opr3, Opr4, Opr5, Opr6, Opr7, Opr8			Opr1, Opr2, Opr3, Opr4, Opr6, Opr7, Opr8		
$N_{S_{opr}}$	8			7		
$N_{S_{opr}}$	8			7		
$S_{opr}[i]$	Opr4, Opr1, Opr7	Opr5, Opr8, Opr1, Opr2	Opr8, Opr7, Opr6, Opr3	Opr4, Opr1, Opr7	Opr8, Opr1, Opr2	Opr8, Opr7, Opr6, Opr3
$S_{sopr}[i]$	Opr4	<u>Opr5</u> , <u>Opr2</u>	Opr6, Opr3	Opr4	<u>Opr2</u>	Opr6, Opr3
$S_{mopr}[i]$	Opr1, Opr7	Opr1, Opr8	Opr7, Opr8	Opr1, Opr7	Opr1, Opr8	Opr7, Opr8
$E_{opr}[i, j]$	0.40, 0.70, 0.75	1.10, 1.20, 1.30, 1.40	0.20, 0.25, 0.40, 0.50	0.40, 0.70, 0.75	1.20, 1.30, 1.40	0.20, 0.25, 0.40, 0.50
$E_{sopr}[i, j]$	0.40	1.10, 1.40	0.40, 0.50	0.40	1.40	0.40, 0.50
$SAM[i]$	1.20	0.60	1.80	1.20	0.60	1.80
$N_{S_{opr}[i]}$	3	1	4	2	1	4
$N_{S_{opr}[i]}$	3	4	4	3	3	4
$N_{S_{sopr}[i]}$	1	<u>2</u>	2	1	<u>1</u>	2
$N_{S_{mopr}[i]}$	2	2	2	2	2	2

Analysis shows that the number of single skilled operators

for the second operation 2 is larger than that of operators really needed, 1. No combination is possible to be obtained under this condition. So it has to be adjusted. Using the approach in A part of Section 3, the optimal number of single skilled operators on Opt 2 is achieved as 1. Thus one of the two available single skilled operators, Opr5 and Opr2, has to be removed. As Opr 2's efficiency is higher than Opr5's, Opr 5 is hence removed from the set. After recalculation, $N_{S_{opr}}[i]$ changes to $\{2, 1, 4\}$. $N_{sopr}[i]$ to $\{3, 3, 4\}$ and $N_{S_{opr}}[i]$ to $\{1, 1, 2\}$. Continue the above steps on all left operations, till all conditions are confirmed to be satisfied. The final result is shown on the right side of Table 1.

After that, run the adjusted operator allocation optimization algorithm to achieve two feasible allocations. The optimal operator allocation is chosen in terms of three indices. Details are shown in Tables 2 and 3.

Table2: Case 1: three indices of the optimal operator allocation

Eff of bottleneck	Std of opeff	total opeff waste
1.15	0.1323	0.45

Table 3: Case 1: the optimal operator allocation

	Opt1	Opt2	Opt3
$S_{opr}[i]$	Opr4, Opr7	Opr2	Opr8, Opr7, Opr6, Opr3

B. Case of Adjustment for Condition of $N_{sopr}[i] < N_{S_{opr}}[i]$

The following experiment will demonstrate with case 2 the approach of how to adjust the condition of $N_{sopr}[i] < N_{S_{opr}}[i]$ and report the corresponding result of the optimal operator allocation.

In this case, before adjustment, S_{opr} , S_{opt} , N_{sopr} and $N_{S_{opr}}$ are the same as Case 1 as shown in table4. Skill matrix and predicted operator efficiency are different. $SAM = \{2.40, 0.50, 1.00\}$, and $N_{S_{opr}}[i]$ is calculated as $\{5, 1, 2\}$. The number of total available operators for each operation $N_{sopr}[i]$ is $\{4, 3, 6\}$ and the number of single skilled operators for each operation $N_{S_{opr}}[i]$ is $\{1, 1, 1\}$. Analysis shows that the number of needed operators for the first operation, 5, is larger than that of all operators available, 4. Using the approach in B part of Section 3, the optimal number of needed operators on Opt 1 is 4. After recalculation, $N_{S_{opr}}[i]$ changes to $\{4, 1, 2\}$. $N_{sopr}[i]$ and $N_{S_{opr}}[i]$ keep unchanged. The system continues to check all left operations, till all conditions are satisfied.

An optimal operator allocation is selected from three available allocations after running the adjusted operator allocation optimization algorithm. Three indices and details of the optimal allocation are shown as follows:

Table 4: Condition adjustment for operator allocation in case 2

$i \backslash j$	Before adjustment			After adjustment		
	Opt1	Opt2	Opt3	Opt1	Opt2	Opt3
S_{opr}	Opr1, Opr2, Opr3, Opr4, Opr5, Opr6, Opr7, Opr8			Opr1, Opr2, Opr3, Opr4, Opr5, Opr6, Opr7, Opr8		
N_{sopr}	8			8		
$N_{S_{opr}}$	8			7		
$S_{opr}[i]$	Opr4, Opr6, Opr5, Opr7	Opr8, Opr1, Opr2,	Opr7, Opr3, Opr2, Opr6, Opr5, Opr8	Opr4, Opr6, Opr5, Opr7	Opr8, Opr1, Opr2,	Opr7, Opr3, Opr2, Opr6, Opr5, Opr8
$S_{sopr}[i]$	Opr4	Opr1	Opr3	Opr4	Opr1	Opr3
$S_{mopr}[i]$	Opr5, Opr6, Opr7	Opr2, Opr8	Opr2, Opr5, Opr6, Opr7, Opr8	Opr5, Opr6, Opr7	Opr2, Opr8	Opr2, Opr5, Opr6, Opr7, Opr8
$E_{opr}[i, j]$	0.20, 0.28, 0.30, 0.32	1.25, 1.30, 1.40	0.58, 0.60, 0.65, 0.65, 0.70, 0.72	0.20, 0.28, 0.30, 0.32	1.25, 1.30, 1.40	0.58, 0.60, 0.65, 0.65, 0.70, 0.72
$E_{sopr}[i, j]$	0.20	1.30	0.60	0.20	1.30	0.60
$SAM[i]$	2.40	0.50	1.00	2.40	0.50	1.00
$N_{S_{opr}}[i]$	5	1	2	4	1	2
$N_{sopr}[i]$	4	3	6	4	3	6
$N_{S_{opr}}[i]$	1	1	1	1	1	1
$N_{S_{opr}}[i]$	3	2	5	3	2	5

Table 5: Case 2: three indices of the optimal operator allocation

Eff of bottleneck	Std of opeff	total opeff waste
1.10	0.1007	0.28

Table 6: Case 2: the optimal operator allocation (Opr 2 is removed in this allocation)

	Opt1	Opt2	Opt3
$S_{opr}[i]$	Opr4, Opr6, Opr5, Opr7	Opr1	Opr7, Opr3

VI. CONCLUSION

Previous work [6] studied the operator allocation under the satisfied condition and this paper discusses the allocation under two unsatisfied conditions. The difference among them is that all operators selected will be allocated to produce for the new order if condition (3) is satisfied, while under conditions (4) and (5), some of them may have to be removed from the total operators list to meet the condition. The common point relies on the operator allocation principle: to achieve the most balanced assembly line whatever the condition is. Among the most balanced lines generated, that with the highest bottleneck efficiency and lowest operator efficiency waste will be taken as the best solution.

ACKNOWLEDGMENT

Thank Research Grant Council of the Hong Kong Special Administrative Region for supporting my project. Special thanks should be given to my supervisors, Dr. Wong, Prof. Fan and Dr. Chan as well. Thanks for their patient instructions and warm help.

REFERENCES

- [1] W.K. Wong, S.Y.S. Leung, and P.Y. Mok, "Development of a genetic optimization approach for balancing an apparel assembly line," *International Journal of Advanced Manufacturing Technology*, 2005, DOI: 10.1007/s00170-004-2350-x, 1-8.
- [2] S. M. Eom, S. M. Lee, E. B. Kim, & C. Somarajan, "A Survey of Decision Support System Applications (1988-1994)," *The Journal of the Operational Research Society*, 49(2), 1998, pp.109-120
- [3] Racine, R., Chen, C., & Swift, F. "The impact of operator efficiency on apparel production planning". *International Journal of Clothing Science and Technology*, 4(2/3), 1992, pp.18-25.
- [4] B.L. Song, W.K. Wong, J. Fan, and S.F. Chan, "Real-time intelligent optimization decision support system for line-balancing control," *The proceedings of the 9th World Multi-conference on Systemics, Cybernetics and Informatics*, Orlando, Florida, USA, 2005, July 10 – 13.
- [5] B.L. Song, W.K. Wong, J. Fan, and S.F. Chan, "Prediction of operator efficiency using time series and artificial neural network techniques," *Computers & Industrial Engineering*, 2006, submitted for publication.
- [6] B.L. Song, W.K. Wong, J. Fan, and S.F. Chan, "A recursive operator allocation approach for assembly line-balancing optimization problem with the consideration of operator efficiency," *Computers & Industrial Engineering*, 2006, accepted for publication.